

Fig. 1

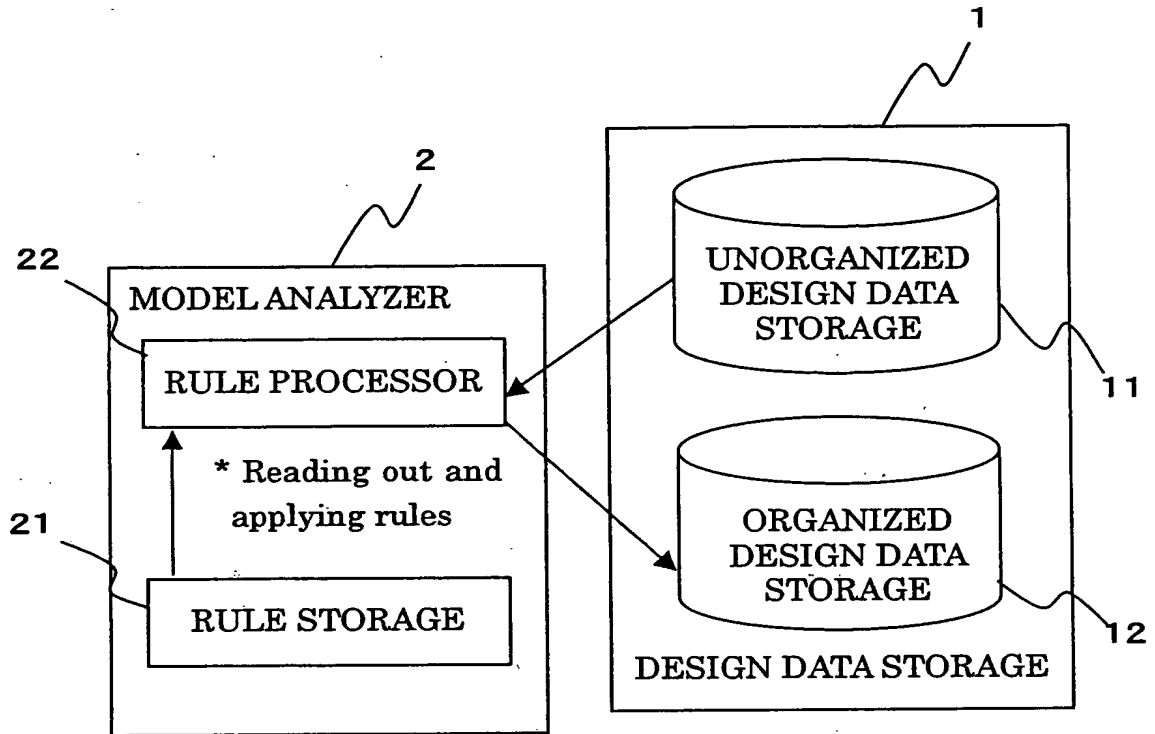


Fig. 2

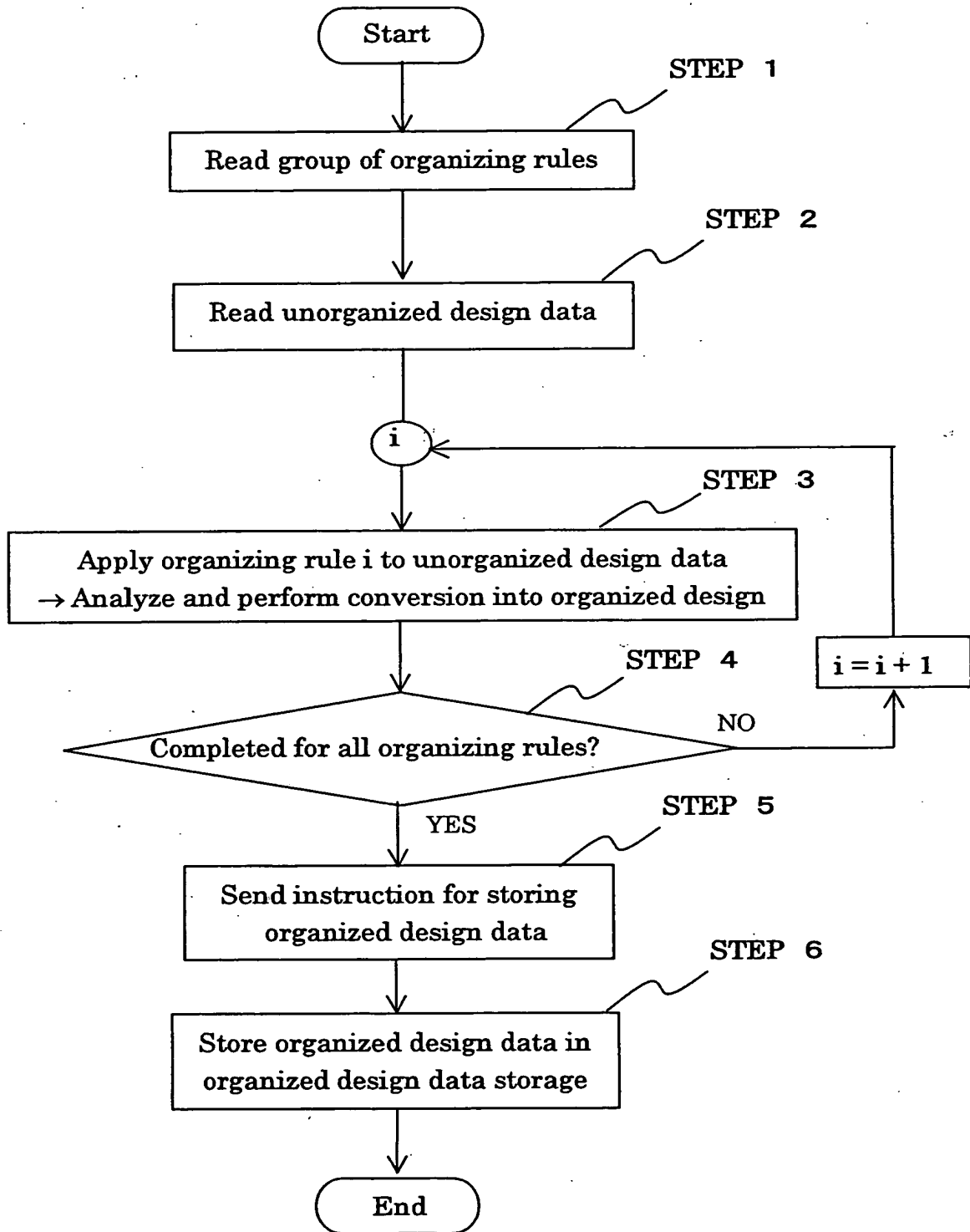
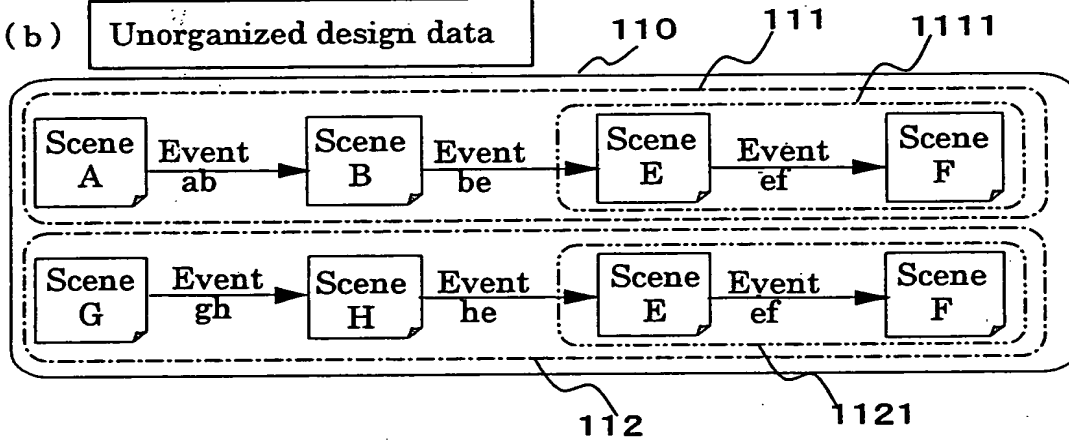


Fig. 3

(a) Organizing rule

Name	Condition	Condition values	Applied processing
Layering of duplicate definitions	Scene sequence segment including n or more elements appears m or more times	n=2 m=2	Cut out target scene sequence segment as one state and perform layering

(b) Unorganized design data



(c) Organized design data

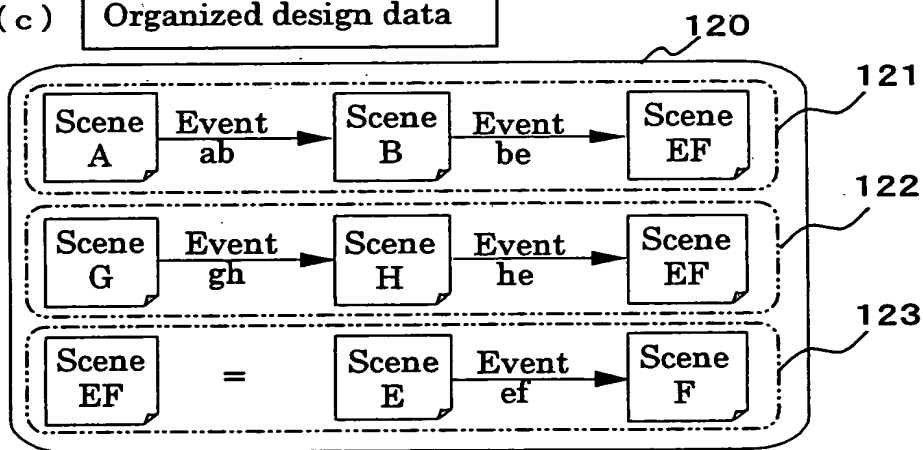


Fig. 4

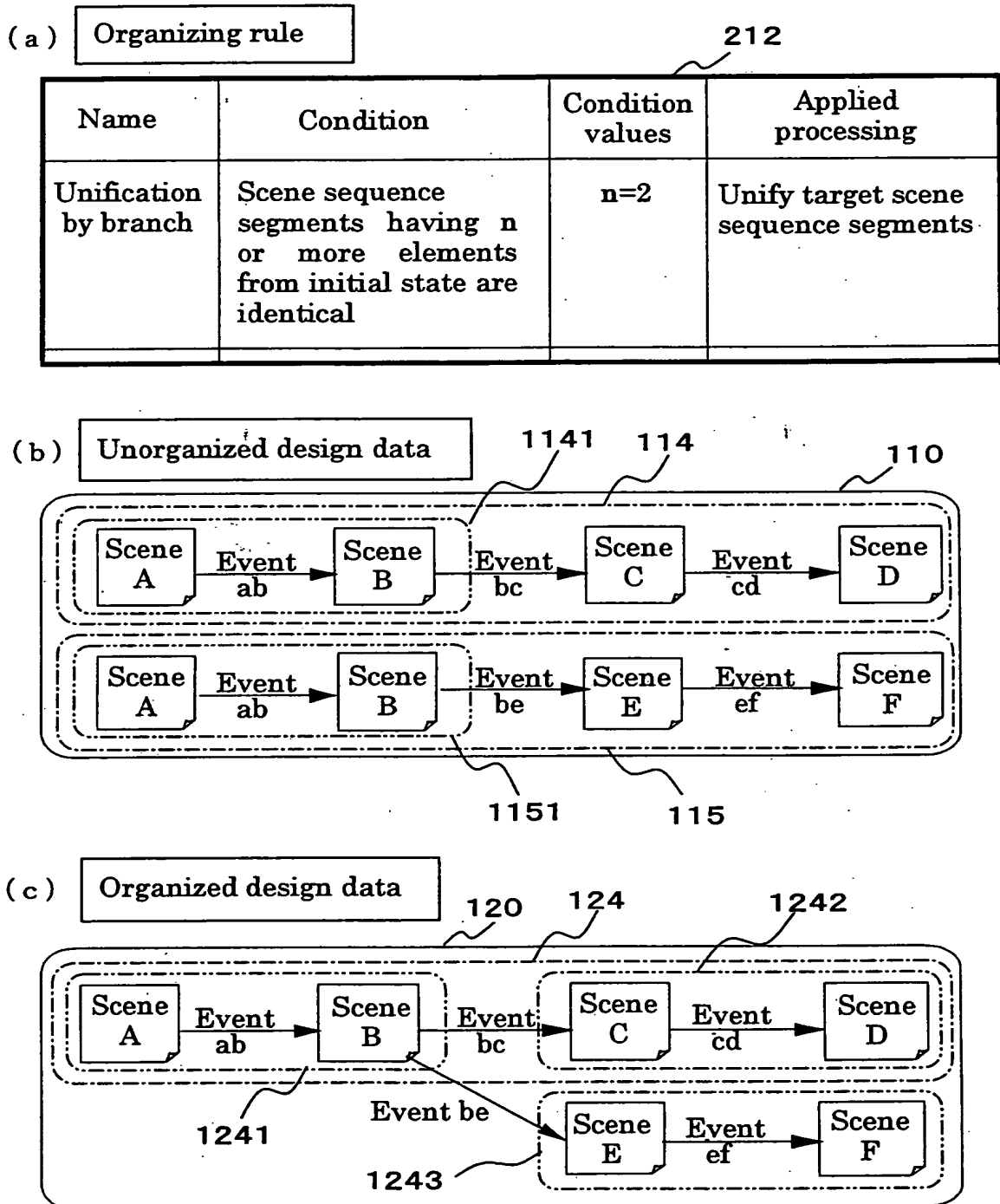


Fig. 5

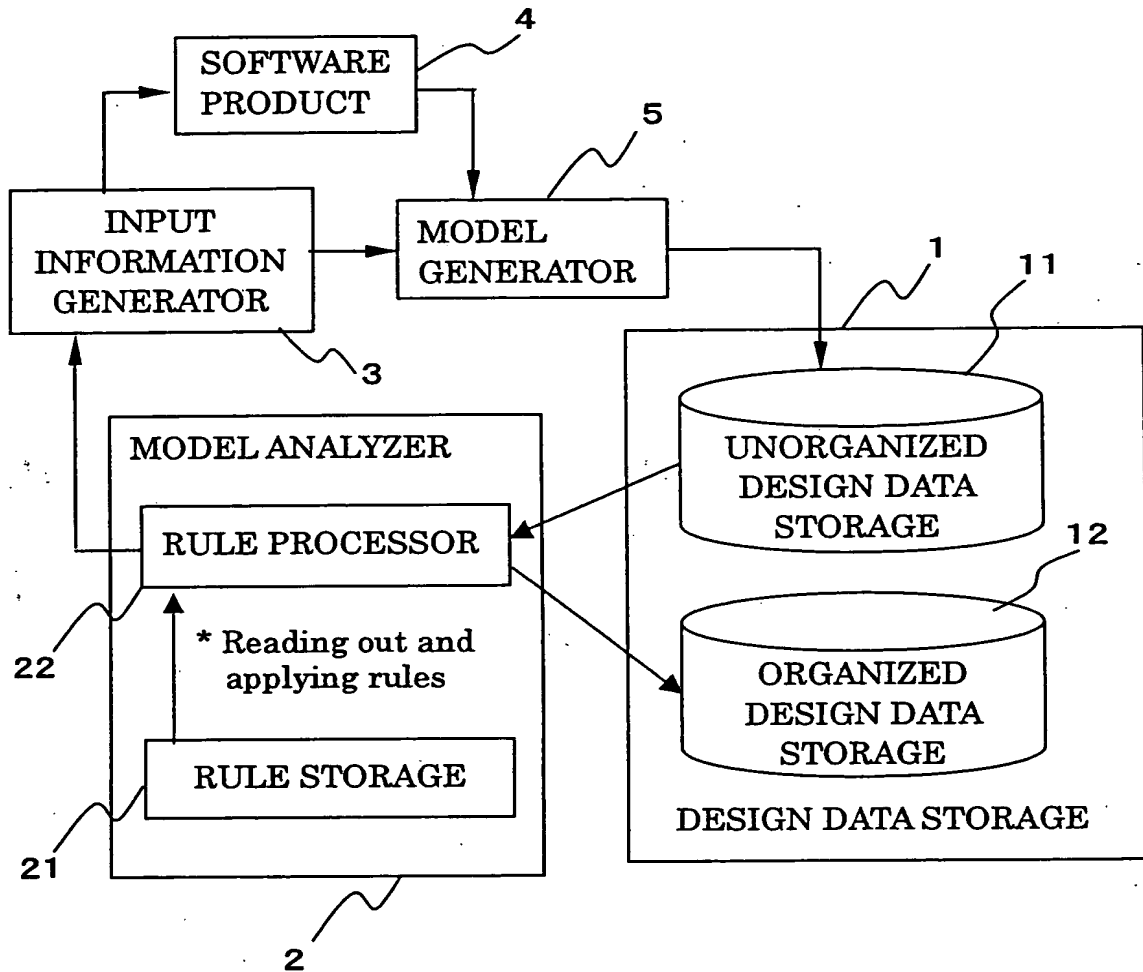


Fig. 6

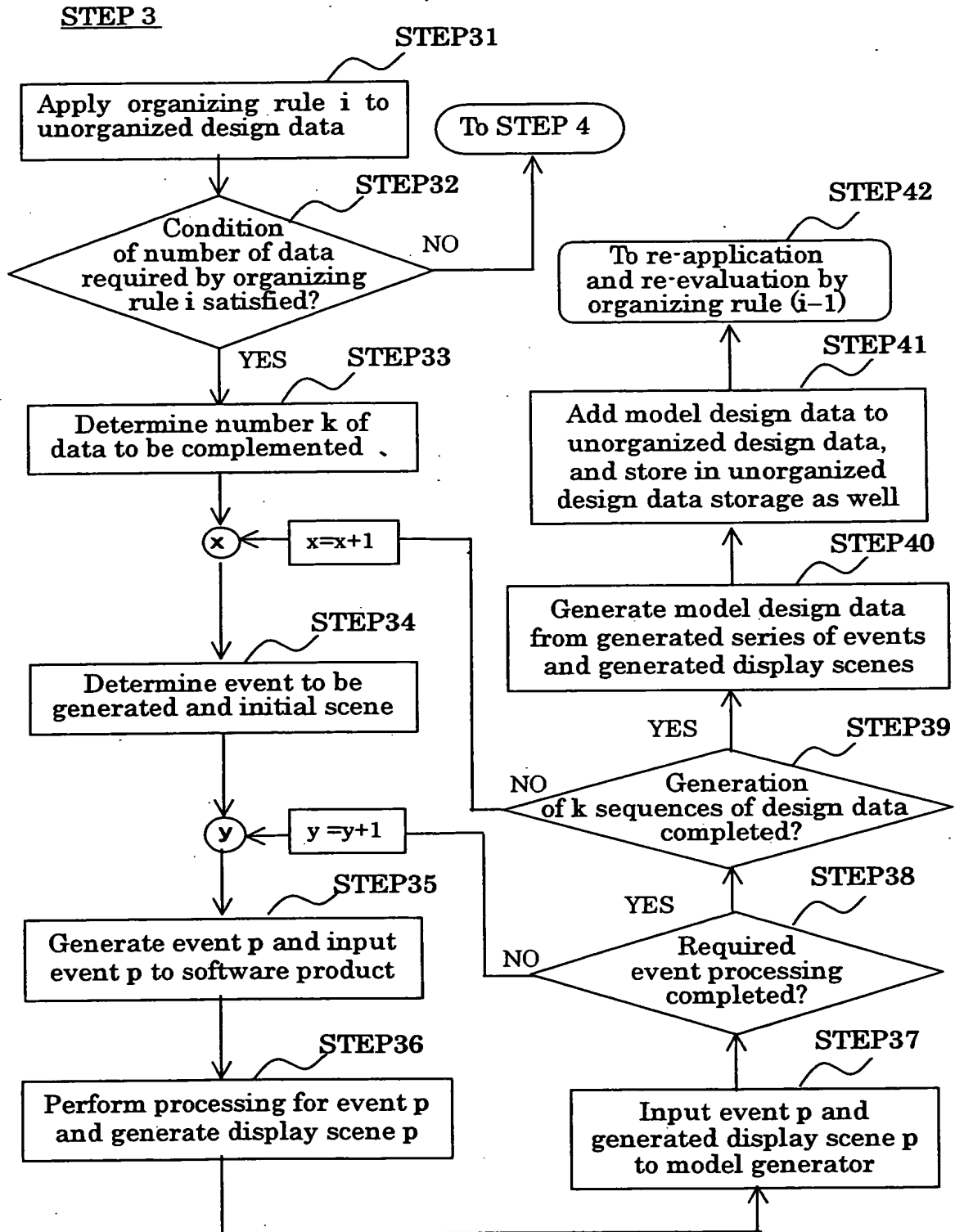


Fig. 7

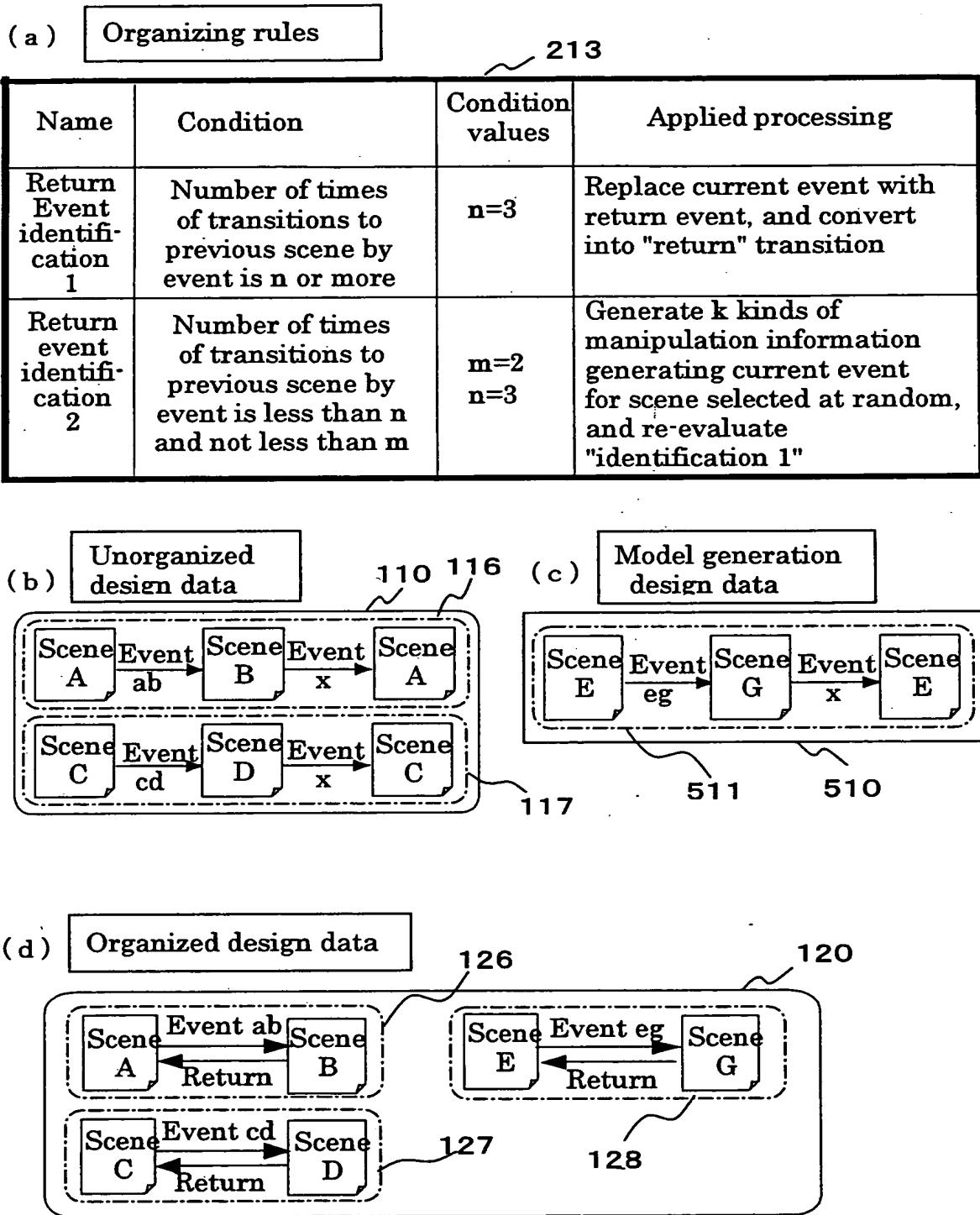


Fig. 8.

(a) **Organizing rules** 214

Name	Condition	Condition values	Applied processing
Intermediate scene unification 1	When scenes included in n or more scene sequences are identical, and transition destination varies according to event in current scene regardless of scene sequences (histories) up to current scene	n=4	Unify current scenes
Intermediate scene unification 2	When scenes included in less than n and not less than m scene sequences are identical, and for less than n scene sequences transition destination varies according to event in current scene regardless of scene sequences up to current scene	m=2 n=4	Input insufficient conditions to input information generator and re-evaluate "Intermediate scene unification 1"

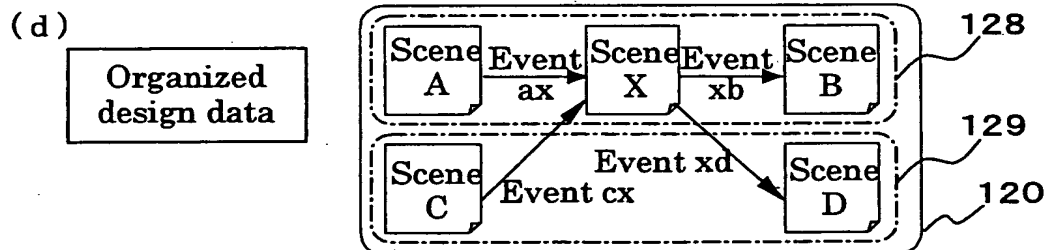
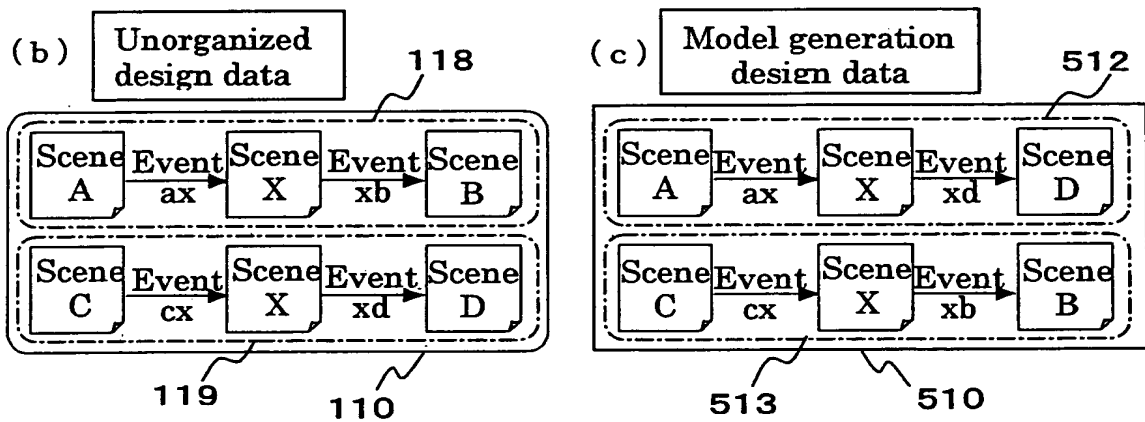


Fig. 9

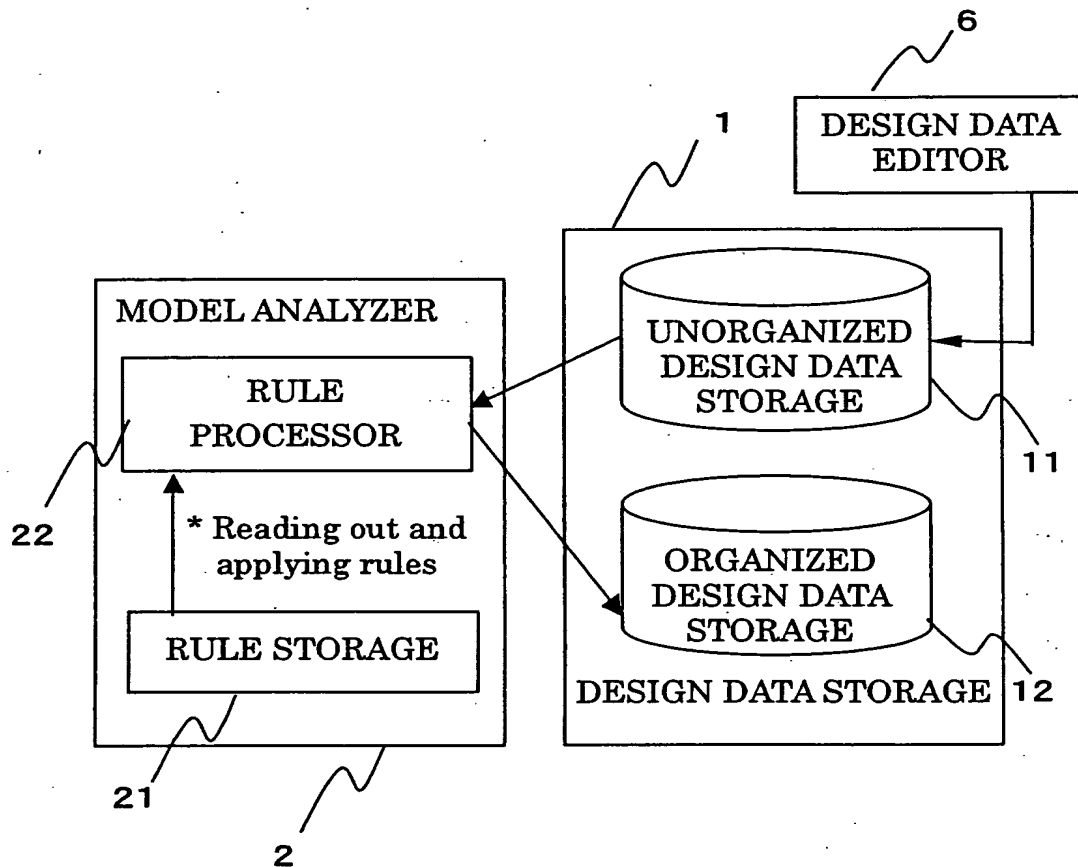


Fig. 10.

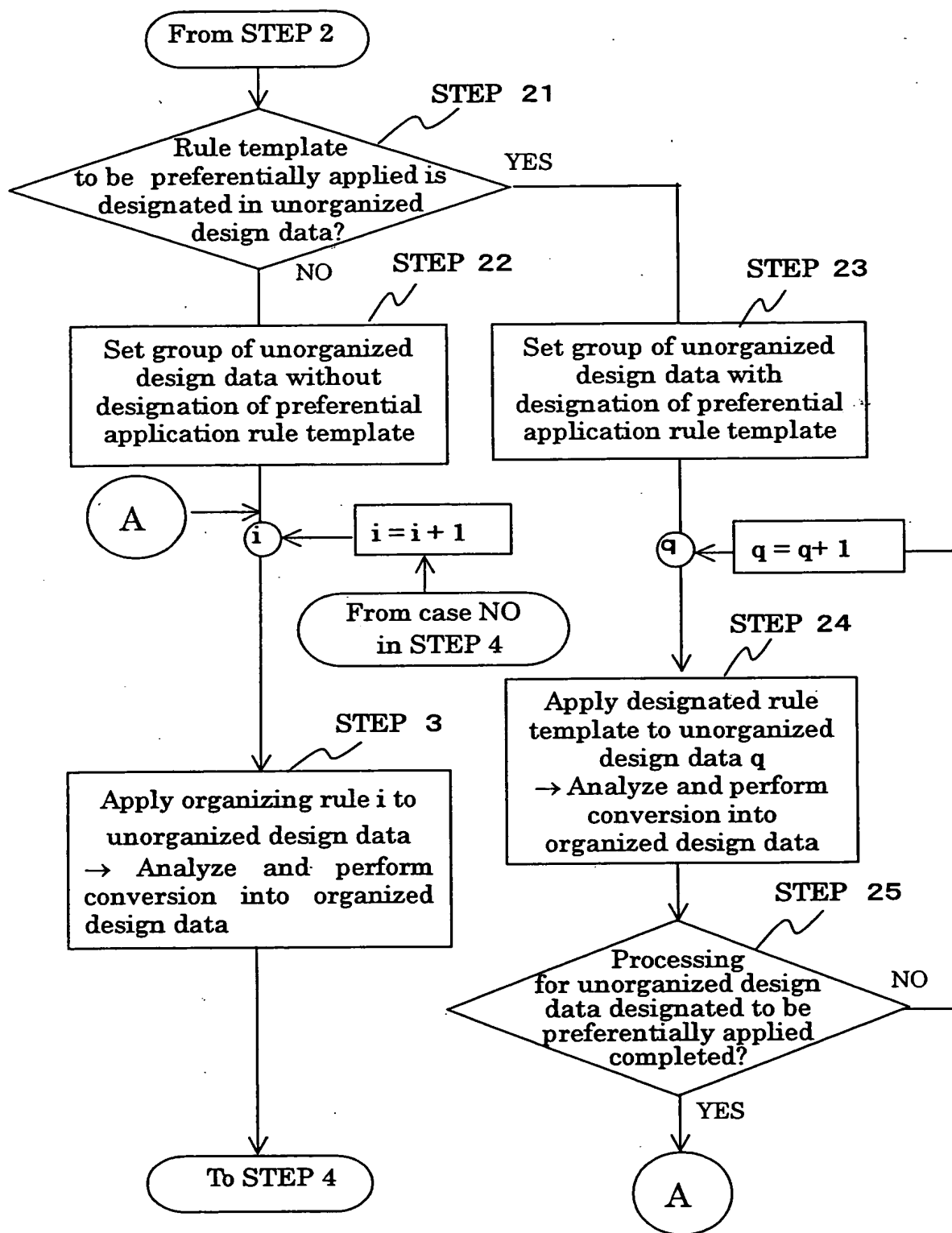


Fig. 11

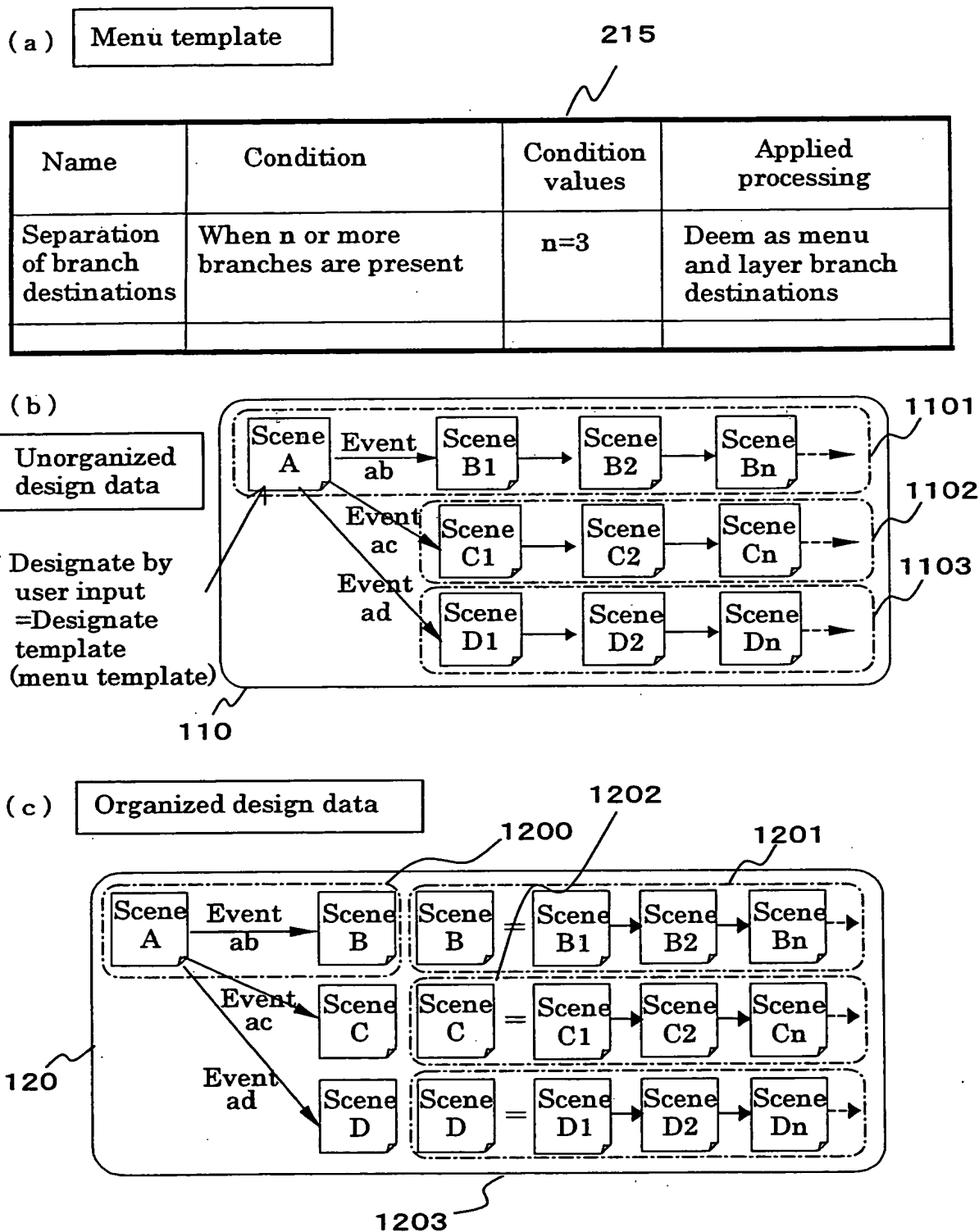


Fig. 12

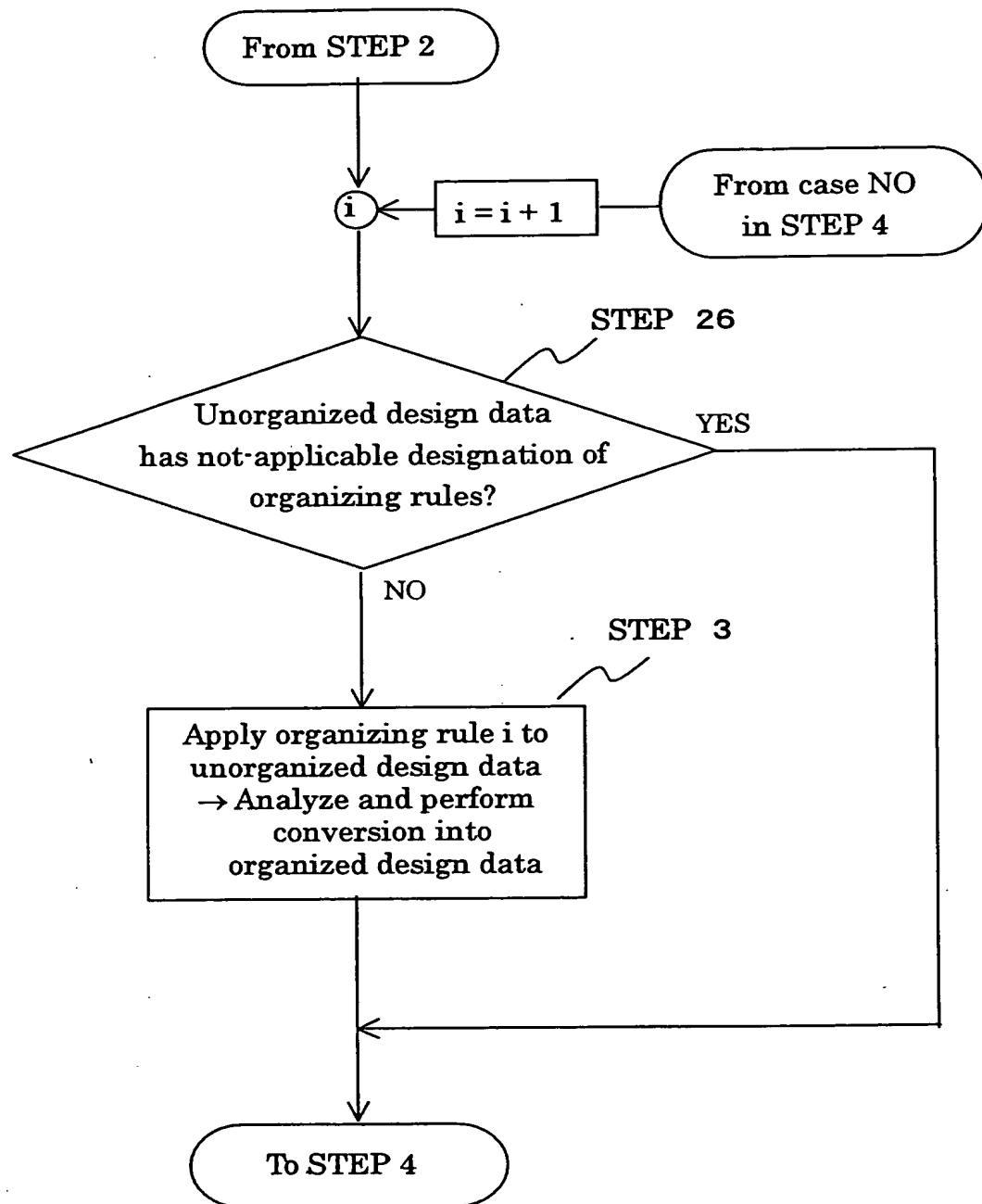
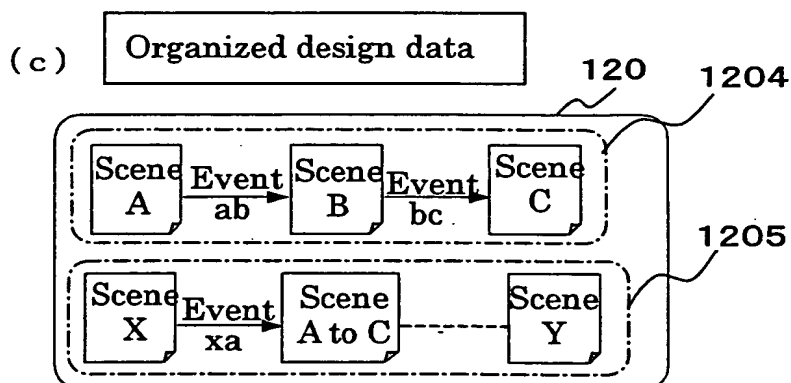
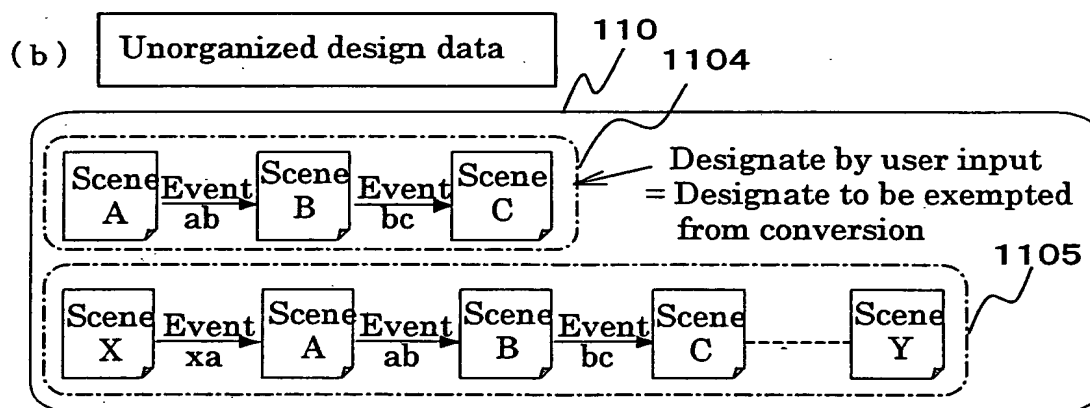


Fig. 13

(a) **Organizing rule** 216

Name	Condition	Condition values	Applied processing
Elimination of duplication	When including entire existing scene sequence		Replace by calling existing scene sequence
* * *			



The diagram illustrates a design data processing system (1) with the following components and data flow:

- Rule Editor (7)**: Provides input to the **Rule Storage (21)**.
- Model Analyzer (22)**: Contains the **Rule Processor (23)** and **Rule Storage (21)**. It receives input from the **Input Information Generator (3)** and the **Rule Editor (7)**. The **Rule Processor (23)** performs the operation: "* Reading out and applying rules".
- Input Information Generator (3)**: Receives input from the **Model Analyzer (22)** and the **Model Generator (5)**. It outputs to the **Software Product (4)**.
- Model Generator (5)**: Receives input from the **Design Data Editor (6)** and the **Design Data Storage (11)**. It outputs to the **Input Information Generator (3)**.
- Design Data Editor (6)**: Provides input to the **Model Generator (5)**.
- Design Data Storage (1)**: Contains two sub-storages:
 - Unorganized Design Data Storage (11)**: Receives input from the **Model Generator (5)** and the **Design Data Editor (6)**.
 - Organized Design Data Storage (12)**: Receives input from the **Unorganized Design Data Storage (11)** and the **Rule Processor (23)**.
- Software Product (4)**: The final output of the system, generated by the **Input Information Generator (3)**.

```
graph TD
    7[Rule Editor] --> 21[Rule Storage]
    21 --> 23[Rule Processor]
    23 --> 3[Input Information Generator]
    3 --> 4[Software Product]
    3 --> 5[Model Generator]
    5 --> 3
    6[Design Data Editor] --> 5
    5 --> 11[Unorganized Design Data Storage]
    6 --> 11
    11 --> 12[Organized Design Data Storage]
    12 --> 23
    23 --> 12
```